WHAT IS CLAIMED IS:

1. A method for classifying received network data comprising: scanning incoming network data using lexical token scanning wherein said network data is treated as a stream of input bytes, said network data being organized into data packets, said lexical scanning resulting in the identification of a data packet as belonging to one of a plurality of classes.

2. The method of claim 1 wherein said scanning includes identifying an arithmetic operation and performing said arithmetic operation.

3. The method of claim 1 wherein said scanning includes identifying a skip operation and in response thereto skipping over one or more subsequent input bytes.

4. The method of claim 1 wherein said lexical scanning includes providing a set of regular expressions, each regular expression having an associated class identifier.

5. The method of claim 1 further including providing a deterministic finite automaton (DFA) comprising plural states, said step of scanning including recognizing data packets using said DFA including transitioning from one state to another.

6. The method of claim 5 wherein said data packets are variable length data packets.

7. The method of claim 5 wherein said DFA is defined by a set of regular expressions.

8. The method of claim 7 further including generating a grammar tree data structure representative of said regular expression, producing a non-deterministic finite automaton (NFA) from said grammar tree data structure, and converting said NFA to produce said DFA.

9. The method of claim 5 wherein some of said states include one or more associated computer instructions and wherein said computer instructions are executed in connection with transitioning to a state.

10. The method of claim 9 wherein some of said states further include a skip instruction.

11. In a network data switching device, a method for classifying data packets comprising steps of:

providing one or more regular expressions, each having an associated class identifier;

receiving plural data packets, each having a length not necessarily equal to one another; and

for each data packet, determining a matching one of said regular expressions that matches said data packet, wherein said each data packet is classified according to the class identifier associated with said matching regular expression.

12. The method of claim 11 wherein said data packets comprise a data stream and said determining includes lexically scanning said data stream.

13. The method of claim 11 wherein said regular expressions are represented by a deterministic finite automaton (DFA).

14. The method of claim 13 wherein said DFA is in compressed form.

15. The method of claim 11 further including compiling said regular expressions to produce said DFA.

16. The method of claim 15 wherein said compiling produces a non-deterministic finite automaton (NFA) as intermediate data structure, said compiling further includes converting said NFA to produce said DFA.

17. The method of claim 16 further including reducing said DFA to a compressed form.

18. The method of claim 11 wherein said data packet comprises plural bytes, and said determining includes detecting an operator indicating a number of bytes to be skipped.

19. The method of claim 18 wherein said number is specified by the value of a current input byte.

20. The method of claim 18 wherein said number is specified in a register.

21. The method of claim 18 wherein said determining further includes detecting an operator indicating a value to be saved in a register.

22. The method of claim 21 wherein said determining further includes detecting an operator indicating a logical or mathematical operation to be performed on the contents of said register.

23. In a data packet receiving and forwarding device, a method for classifying received data packets comprising a stream of data, said method comprising steps of:

receiving a description of classification rules in a classification language;

compiling said classification language to produce a deterministic finite automaton (DFA) comprising plural states;

configuring a programmable hardware packet classifier with said DFA; and

scanning said data stream with said hardware packet classifier to classify said received data packets.

24. The method of claim 23 wherein said compiling includes associating arithmetic and logic instructions with some of said states.

25. The method of claim 23 wherein said classification language includes regular expressions.

26. The method of claim 25 wherein said regular expressions include arithmetic and logic operations.

27. The method of claim 26 wherein said regular expressions further include skip operations.

28. The method of claim 27 wherein said regular expressions further include data storage operations.

29. The method of claim 23 wherein said DFA is in compressed format.

30. The method of claim 23 further including:

receiving a second description of classification rules in a classification language;

compiling said second classification language to produce a second DFA;

configuring a programmable hardware packet classifier with said second DFA; and

applying said data stream to said hardware packet classifier to classify said received data packets,

wherein said data packets are classified according to said second classification rules, thereby facilitating changing packetizing policies in said data packet routing device.

31. A network data packet classifier comprising:

an input port for receiving network data packets comprising a stream of data;

a memory assemblage configured with data representing a deterministic finite automaton (DFA), said DFA representing plural regular expressions; and

decompression logic operatively coupled to said memory assemblage and configured to scan said stream of data with said DFA to find a matching one of said regular expressions,

said regular expressions having corresponding class identifiers,

wherein each of said network data packets is associated with the class identifier of said regular expression that matches it.

32. The classifier of claim 31 wherein some of said regular expressions include arithmetic instructions and logic instructions, said memory assemblage further

configured to contain said instructions, the classifier further including an arithmetic logic unit operatively coupled to said decompression logic and configured to execute said instructions.

33. The classifier of claim 32 further including at least one register operatively coupled to said arithmetic logic unit, said arithmetic logic unit further configured to store data into said register in response to a save instruction.

34. The classifier of claim 32 further including skip logic operatively coupled to said logic component and configured to skip over an amount of data in response a skip instruction.

35. The classifier of claim 31 wherein said network data packets can vary from one packet to another.

36. The classifier of claim 31 wherein said DFA is in compressed form.

37. The classifier of claim 36 wherein said DFA comprises plural non-default states and plural default states, and said memory assemblage comprises a base memory, a next-state memory, and a default-state memory; said base memory configured to contain address locations of said next-state memory, said next-state memory representing all of said non-default states, said default-state memory representing all of said default states.

38. The classifier of claim 37 wherein said memories are random access memories.

39. The classifier of claim 37 wherein said memories are read-only memories.

40. A network data packet classifier comprising:

an input configured to provide a data packet comprising a stream of data;

a first system of memory configured with data representing a deterministic finite automaton (DFA), said DFA comprising plural states including an initial state and plural terminating states;

a system of logic circuits operatively coupled to said first system of memory and to said input, and configured to lexically scan said data stream with said DFA to produce a reached terminating state; and

a second system of memory configured with data representing a class index corresponding to each of said terminating states and configured to output a class index in response to the production of said reached terminating state.

41. The classifier of claim 40 further including a third system of memory configured to contain current state information for plural input channels, said system of logic circuits operatively coupled to said third system of memory to initialize said DFA in accordance with current state information corresponding to the input channel associated with said data packet.

42. The classifier of claim 40 wherein some of said states have one or more associated instructions, the classifier further including an arithmetic logic unit operatively coupled to said system of logic circuits and configured to execute said instructions.

43. The classifier of claim 42 further including at least one register operatively coupled to said arithmetic logic unit, said arithmetic logic unit further configured to store data into said register in response to a save instruction.

44. The classifier of claim 42 further including skip logic operatively coupled to said logic component and configured to skip over an amount of data in response a skip instruction.

45. The classifier of claim 40 wherein said stream of data is a stream of bytes.

46. The classifier of claim 40 wherein said data packets vary from one packet to another.

47. A network packet classifier comprising:

means for receiving an incoming network packet;

means for classifying said network packet by matching the pattern of its constituent data against plural regular expressions, each regular expression having a corresponding class identifier; and

means for outputting a class identifier of the regular expression which matches said network packet.

48. The classifier of claim 47 wherein said means for classifying includes a memory component configured with data to represent a deterministic finite automaton (DFA).

49. The classifier of claim 48 wherein said means for outputting includes a second memory component configured with said class identifiers.

50. The classifier of claim 47 wherein said regular expressions include arithmetic specifiers and said means for classifying includes an arithmetic logic unit configured to perform operations in accordance with said arithmetic specifiers.

51. A network packet classifier comprising:

a dual-ported memory component;

first classification logic operatively coupled to a first port of said dual-ported memory component and having a first input for receiving a data stream; and

second classification logic operatively coupled to a second port of said dual-ported memory component and having a second input for receiving a data stream,

said memory component configured to contain a deterministic finite automaton (DFA) comprising plural states,

said DFA representing plural regular expressions for matching data packets,

said first and second classification logic each configured to scan its associated data stream using said DFA to identify data packets contained therein and to classify identified data packets.

52. The classifier of claim 51 wherein said data packets are characterized in being variable in length.

53. The classifier of claim 51 wherein said regular expressions include arithmetic and logic operators.

54. The classifier of claim 51 wherein said regular expressions include a skip operator.